Here are some of the most commonly asked **basic AWS DevOps Engineer interview questions** for someone with **4+ years of experience**. At your experience level, interviewers usually expect practical answers rather than textbook definitions.

My self Rohith, currently working in Mphasis limited with 4+ years of experience as Devops engineer

& my day to day role is to support interface production environment, I manage the junkins pipelines starting to end by using Git for the version control system & Docker for containerization & to manage the cloud instances we use terraform which provides the plateform for applications & for container orchirazation we use kubernates & also worked on AWS service like s3(simple storage service), bucket for storage, ec2 instance for scalable the servers & to provide the access to the bank employee we use identity access management to create groups & I have created rules also I manage the linux interface there & also based on day to day work I monitor cpu, disc utilization along with that I have knowledge of n\w (explain ur work)

1. Tell me about yourself.

Answer: Give a 1–2 minute introduction covering:

- Total experience
- Current role
- AWS services you've worked on
- DevOps tools (Git, Jenkins, Docker, Kubernetes, Terraform, Ansible)
- Your key responsibilities

---

2. What is DevOps?

Answer:
DevOps is a culture and set of practices that bring together development and operations teams to automate software delivery, improve collaboration, and enable faster, more reliable releases using CI/CD and Infrastructure as Code.

---

3. What is CI/CD?

CI (Continuous Integration)

- Developers frequently merge code.
- Automated builds and testing are triggered.

CD (Continuous Delivery/Deployment)

- Code is automatically deployed to testing or production environments.

Tools: Jenkins, GitHub Actions, GitLab CI, AWS CodePipeline.

---

4. Explain your CI/CD pipeline.

Example:

1. Developer pushes code to Git.
2. Jenkins detects the change.
3. Maven builds the application.
4. Unit tests are executed.
5. Docker image is created.
6. Image is pushed to Docker Hub or Amazon ECR.
7. Kubernetes deploys the new version.
8. Monitoring through CloudWatch.

---

5. What AWS services have you worked on?

Example:

- EC2
- VPC
- IAM
- S3
- Route 53
- CloudWatch
- Auto Scaling
- Load Balancer
- RDS
- ECR
- EKS
- CloudFormation/Terraform

---

6. What is EC2?

Amazon EC2 is a virtual server in AWS used to host applications with scalable compute capacity.

---

7. Difference between Security Group and NACL

Security Group	NACL
Instance level	Subnet level
Stateful	Stateless
Allow rules only	Allow and Deny rules
Default allows return traffic	Must explicitly allow return traffic

---

8. What is IAM?

IAM (Identity and Access Management) controls access to AWS resources using:

- Users
- Groups
- Roles
- Policies

---

9. Difference between IAM Role and IAM User

IAM User

- Permanent identity
- Login credentials

IAM Role

- Temporary permissions
- Used by EC2, Lambda, EKS, and cross-account access

---

10. What is S3?

Amazon S3 is an object storage service used for:

- Backups
- Static website hosting
- Log storage
- Artifacts
- Media files

---

11. What is Auto Scaling?

Automatically increases or decreases EC2 instances based on metrics like CPU utilization.

---

12. What is an Application Load Balancer?

An ALB distributes incoming HTTP/HTTPS traffic across multiple EC2 instances or containers to improve availability and scalability.

---

13. What is Terraform?

Terraform is an Infrastructure as Code (IaC) tool that provisions cloud infrastructure using declarative configuration files.

Example:

```
terraform init
terraform plan
terraform apply
terraform destroy
```

---

14. Why Terraform instead of CloudFormation?

Terraform:

- Multi-cloud support
- Easier syntax
- Better module reusability

CloudFormation:

- AWS-only
- Native AWS integration

---

15. What is Docker?

Docker packages applications and their dependencies into containers, ensuring consistent behavior across environments.

---

16. Difference between Docker Image and Container

Image

- Read-only template

Container

- Running instance of an image

---

17. What is Kubernetes?

Kubernetes is a container orchestration platform that automates deployment, scaling, and management of containerized applications.

---

18. What is a Pod?

A Pod is the smallest deployable unit in Kubernetes. It can contain one or more containers.

---

19. What is Jenkins?

Jenkins is an automation server used to build, test, and deploy applications as part of a CI/CD pipeline.

---

20. What is Ansible?

Ansible is a configuration management and automation tool that uses YAML playbooks and communicates over SSH without requiring agents.

---

21. What is Git?

Git is a distributed version control system used to track code changes and enable collaboration.

Common commands:

```
git clone
git status
git add
git commit
git push
git pull
git merge
```

---

22. What is Maven?

Maven is a Java build automation tool used to:

- Download dependencies
- Compile code
- Run tests
- Package applications

---

23. What monitoring tools have you used?

- Amazon CloudWatch
- Prometheus
- Grafana
- ELK Stack (if applicable)

---

24. How do you troubleshoot a failed Jenkins build?

1. Check console output.
2. Verify Git checkout.
3. Check Maven build errors.
4. Verify credentials.
5. Ensure agents are online.
6. Review deployment logs.
7. Fix the issue and rerun the build.

---

25. What are your daily responsibilities?

Example:

- Managing AWS infrastructure
- Creating CI/CD pipelines
- Writing Terraform code
- Managing Docker containers
- Deploying to Kubernetes
- Monitoring applications
- Troubleshooting production issues
- Collaborating with developers
- Automating deployments
- Managing IAM permissions

Questions often asked to 4+ year AWS DevOps engineers

- Explain your end-to-end CI/CD pipeline.
- How do you deploy an application to Kubernetes?
- How do you perform a rolling deployment?
- How do you roll back a failed deployment?
- How do you secure AWS resources?
- Explain your Terraform project.
- Describe a production issue you resolved.
- How do you monitor applications in AWS?
- Explain blue-green and canary deployments.
- How do you optimize AWS costs?

These questions form the core of many AWS DevOps interviews for professionals with 4–5 years of experience. Being able to answer them with examples from your own projects will significantly strengthen your interview performance.